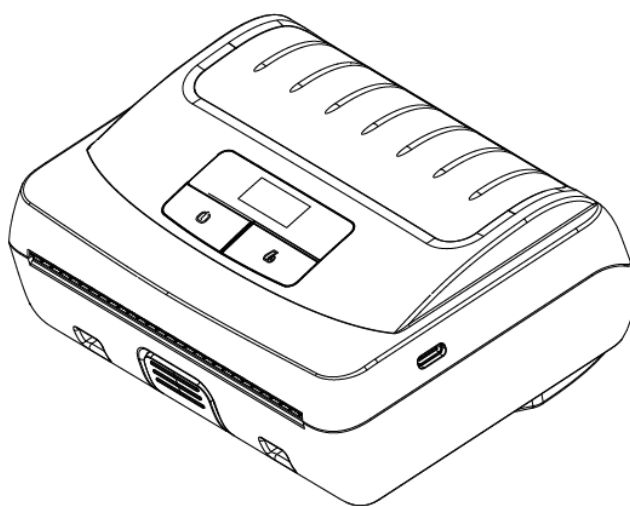


HM-A400

CPCL

编程手册



(Rev.1.0)

目 录

1.前言	1
2.打印机命令	2
打印机命令	2
PRINT 命令	3
END 命令	3
ABORT 命令	3
FORM 命令	3
UNITS 命令	4
CONTRAST 命令	4
TONE 命令	5
使用注释	5
3.文本	6
TEXT 命令	6
TEXT 串联命令（CONCAT 和 VCONCAT）	6
COUNT 命令	7
SETMAG 命令	8
MULTILINE(ML)命令	9
FONT-GROUP（FG）命令	10
4.条码指令	11
BARCODE 命令	11
BARCODE-TEXT 命令	12
PDF417（便携式数据文件）	12
QR Code	13
5.图形	16
LINE 命令	16
INVERSE-LINE 命令	17
GRAPHICS 命令	18
PCX 命令	19
PATTERN 命令	20
6.高级命令	21
PAGE-WIDTH 命令	21
PAGE-HEIGHT 命令	22
PREFEED 命令	22
PACE 命令	22
WAIT 命令	23
PRESENT-AT 命令	23
SPEED 命令	24
SETSP 命令	25
AUTO-PACE 命令	25
ON-FEED 命令	26
COUNTRY/CODE PAGE 命令	27
DEFINE FORMAT	28

USE-FORMAT	28
BEEP 命令	29
7.行式打印模式.....	30
SETLP 命令	30
SETLF 命令	30
使用 X 和 Y 坐标移动	31
LMARGIN 命令	31
SETBOLD 命令	31
LP-ORIENT 命令.....	31
SETFF 命令	32
8.高级实用工具.....	33
TIMEOUT 实用工具	33
CHECKSUM 实用工具.....	33
DIR 实用工具.....	33
DEL 实用工具.....	34
DEFINE-FILE (DF) 实用工具.....	34
TYPE 实用工具.....	35
GET-TIME 实用工具.....	35
GET-DATE 实用工具.....	35

1.前言

本手册详细介绍了 CPCL 语言中的各种命令,通过这些命令,编程人员可以利用打印机中的内置文本、图形、条码打印和通信等功能。文中通篇使用了以下符号约定:

{ } 必填项

[] 可选项

() 缩写命令

{ } 文字项

空格字符用于分隔命令行中的各个字段。

许多命令都随附了命令使用示例。在每个示例中的“Input”一词后面显示的是命令集,后接打印机处理这些命令后生成的打印输出结果 (“Output”) 示例。

2.打印机命令

标签文件通常以“!”字符作为开头，后接“x”偏置参数、“x”和“y”轴分辨率、标签长度以及要打印的标签数量。包含这些参数的行称为命令起始行。

任何情况下，标签文件都是以命令起始行开头，以“PRINT”命令结尾。用于构建具体标签的命令置于这两项命令之间。

空格字符用于分隔命令行中的各个字段。

备注：命令会话中的每一行都必须以回车和换行两种字符结尾。所有打印机命令必须采用大写字符。

打印机命令

格式：

`<!> {offset} <200> <200> {height} {qty}`

其中：

`<!>`：使用“!”作为控制会话的起始字符。

`{offset}`：整个标签的横向偏置。此值可以使所有域以指定的单位数量进行横向偏置。

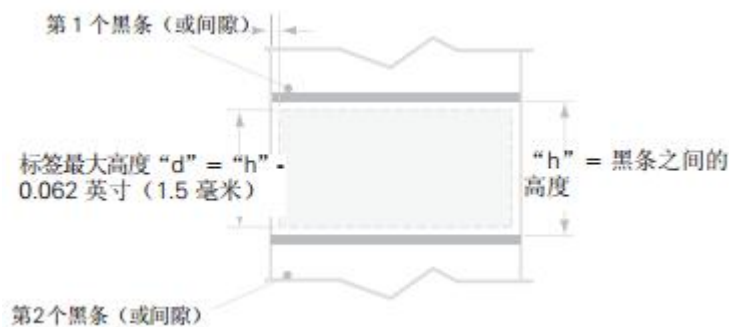
`<200>`：横向分辨率（以点/英寸为单位）。

`<200>`：纵向分辨率（以点/英寸为单位）。

`{height}`：标签的最大高度。

标签最大高度的计算方法是，先测出从第 1 个黑条（或标签间隙）底部到下一个黑条（或标签间隙）顶部之间的距离。然后从中减去 1/16 英寸（1.5 毫米），所得结果即最大高度。（以点为单位时：对于 203 d.p.i 打印机，减去 12 点；对于 306 d.p.i. 打印机，减去 18 点）

`{qty}`：要打印的标签数量。最大值 = 1024。



打印机命令示例

输入

```
! 0 200 200 210 1
TEXT 4 0 30 40 Hello World
FORM
PRINT
```

输出

Hello World

PRINT 命令

PRINT 命令作为整个命令集的结束命令，将会启动文件打印。在任何情况下（行式打印模式除外），这项命令都必须是最后一条命令。执行 PRINT 命令时，打印机将从控制会话中退出。确保使用回车和换行字符结束此项及所有命令。

格式：

{command}

其中：

{command}: PRINT

END 命令

END 命令可以正常终止一项命令，也可在不打印的情况下执行一项命令。

格式： END

{command}

其中：

{command}: END

示例：

REM 发送一个空白标签

! 0 200 200 240 1\r\n

PAGE-WIDTH 240\r\n

BOX 0 0 200 200 10\r\n

BOX 50 50 220 220 10\r\n

END\r\n

ABORT 命令

ABORT 命令可以在不打印的情况下终止当前的控制会话。

格式：

{command}

其中：

{command}: ABORT

FORM 命令

FORM 命令可以指示打印机在一页打印结束后切换至下一页顶部。

格式：

{command}

其中：

{command}: FORM

在下例中，打印机将在打印标签后执行换页操作。有关设置执行 FORM 命令时打印机操作的信息，请参阅设计收据和列表一节中的 SETFF（设置换页）命令。

示例

输入：

! 0 200 200 3 1

IN-CENTIMETERS

CENTER

TEXT 4 1 0 5 Hello World

FORM

PRINT

UNITS 命令

UNITS 命令可以针对控制会话中的所有后续命令字段指定度量系统。输入所有控制命令的坐标、宽度和高度时，均可精确到四位小数位。通过在控制会话的第一行后紧跟单位命令，还可将指定的度量系统应用于偏置和高度字段。在单位命令发出之前，打印机度量系统默认以点为单位。

格式：

{command}

其中：

{command}： 从下面选择一项：

IN-INCHES 度量单位为英寸。

IN-CENTIMETERS 度量单位为厘米。

IN-MILLIMETERS 度量单位为毫米。

IN-DOTS 度量单位为点。

UNITS 命令示例

输入 1：

! 0.3937 200 200 1 1

IN-INCHES

T 4 0 0 0 1 cm = 0.3937"

IN-DOTS

T 4 0 0 48 1 mm = 8 dots

B 128 1 1 48 16 112 UNITS

T 4 0 48 160 UNITS

FORM

PRINT

输出 1：（如右图）

输入 2

! 0 200 200 2.54 1

IN-CENTIMETERS

T 4 0 1 0 1" = 2.54 cm

IN-MILLIMETERS

T 4 0 0 6 203 dots = 25.4 mm

B 128 0.125 1 6 12 14 UNITS

T 4 0 16 20 UNITS

FORM

PRINT

输出 2

1 cm = 0.3937"
1 mm = 8 dots

UNITS

1" = 2.54 cm
203 dots = 25.4 mm

UNITS

CONTRAST 命令

CONTRAST 命令用于指定整个标签的打印浓度。打印输出最浅的是对比度 0，打印机在打开电源时默认为对比度 0。必须为每个标签文件指定对比度。

备注：为了最大限度地提高打印效率，请尽可能使用最低对比度打印。

格式：

{command} {level}

其中:

{command}: CONTRAST

{level}: 对比度级别

0 = 默认

1 = 中度

2 = 浓

3 = 很浓

TONE命令

TONE 命令可以替代 CONTRAST 命令来指定所有标签的打印浓度。打印输出最小的色调级别为-99。最大的色调级别为 200。打印机开机时默认色调级别为 0。在所有打印任务发生更改之前，色调级别设置仍有效。TONE 命令和 CONTRAST 命令不能彼此组合使用。

格式:

{command} {level}

其中:

{command}: TONE

{level}: 从-99 到 200 选择一个值。

对比度相对于色调级别:

对比度 0 = 色调 0 对比度 1 = 色调 100

对比度 2 = 色调 200 对比度 3 = 无对应值

使用注释

注释可以添加在命令会话第一行和“PRINT”命令之间。

在文件中添加注释时，需要将“;”字符置入第一列，以此作为注释行的起始部分。“;”字符与行末尾的所有其他文本都将被忽略。CONCAT 与 ENDCONCAT 命令之间不可添加注释。

注释示例

输入:

! 0 200 200 25 1

IN-MILLIMETERS

JOURNAL

; Center justify text

CENTER

; Print the words ' A COMMENT'

TEXT 5 1 0 5 A COMMENT

; Print the label and go to top of next form

FORM

PRINT

输出:

A COMMENT

3.文本

TEXT 命令

TEXT 命令用于在标签上添加文本。这项命令及其各衍生命令可以控制使用的具体字体号和大小、标签上文本的位置以及文本的方向。标准常驻字体能够以 90 度的增量旋转，如下例所示。

格式：

{command} {font} {size} {x} {y} {data}

其中：

{command}：从下面选择一项：

{command}	效果
TEXT （或 T）	横向打印文本。
VTEXT （或 VT）	逆时针旋转 90 度，纵向打印文本。
TEXT90 （或 T90）	（同 VTEXT。）
TEXT180 （或 T180）	逆时针旋转 80 度，反转打印文本。
TEXT270 （或 T270）	逆时针旋转 270 度，纵向打印文本。

{font}： 字体名称/编号。

{size}： 字体的大小标识。

{x}： 横向起始位置。

{y}： 纵向起始位置。

{data}： 要打印的文本。

示例

输入：

```
! 0 200 200 210 1
TEXT 4 0 200 100 TEXT
TEXT90 4 0 200 100 T90
TEXT180 4 0 200 100 T180
TEXT270 4 0 200 100 T270
FORM
PRINT
```

输出：



TEXT 串联命令（CONCAT 和 VCONCAT）

使用文本串联，可以为字符串分配不同的字符样式，在同一文本行上使用统一间距进行打印。这项命令可以与可缩放字体组合使用。

格式：

{command} {x} {y} {font} {size} {offset} {data} ""{font} {size} {offset} {data} <ENDCONCAT>

其中：

{command}：从下面选择一项：

CONCAT： 横向串联。

VCONCAT： 纵向串联。

{x}: 横向起始位置。
 {y}: 纵向起始位置。
 {font}: 字体名称/编号。
 {size}: 字体的大小标识。
 {offset}: 文本相对起始位置的偏置单位值。用于对齐单个文本字符串或创建上标/下标字符。
 {data}: 要打印的文本。
 <ENDCONCAT>: 终止串联。

文本串联示例

输入:

```
! 0 200 200 210 1
```

```
CONCAT 75 75
```

```
4 2 5 $
```

```
4 3 0 12
```

```
4 2 5 34
```

```
ENDCONCAT
```

```
FORM
```

```
PRINT
```

输出:



COUNT 命令

COUNT 命令可以用于打印多个标签，其中条码中编码的数字文本域或数字数据将针对每个标签依次递增或者递减。TEXT/BARCODE命令字符串必须包含此数字数据，将其作为字符串的最后若干字符。数字数据部分最多可以包含20个字符，且可以以‘-’符号作为前缀。增加或减少数字数据时不能以‘0’为增量或减量。前导零将予以保留。一个标签文件中最多可使用三个COUNT命令。

递增/递减的数字数据包含在TEXT或BARCODE命令中，后面紧跟COUNT命令。

格式:

```
{command} {numeric value}
```

其中:

```
{command}: COUNT
```

{numeric value}: 任何整数值都不能超过20个字符。如果希望减小TEXT/BARCODE值，则可以在值前添加‘-’符号。输出结果中将保留前导零。

COUNT 命令示例

输入:

```
! 0 200 200 210 3
```

```
; Print 3 labels
```

```
CENTER
```

```
TEXT 4 0 0 50 TESTING 001
```

```
COUNT 1
```

```
TEXT 7 0 0 100 Barcode Value is 123456789
```

```
COUNT -10
```

```
BARCODE 128 1 1 50 0 130 123456789
```

```
COUNT -10
```

```
FORM
```

```
PRINT
```

输出：



SETMAG 命令

SETMAG 命令可将常驻字体放大指定的放大倍数。

格式：

{command} {w} {h}

其中：

{command}: SETMAG

{w}: 字体的宽度放大倍数。有效放大倍数为 1 到 16。

{h}: 字体的高度放大倍数。有效放大倍数为 1 到 16。

备注：SETMAG 命令在标签打印后仍保持有效。这意味着要打印的下一标签将使用最近设置的 SETMAG 值。要取消 SETMAG 值并使打印机可以使用默认字体大小，请使用 SETMAG 命令，且放大倍数为 0,0。

SETMAG 命令示例

输入：

```
! 0 200 200 210 1
CENTER
SETMAG 1 1
TEXT 0 0 0 10 Font 0-0 at SETMAG 1 1
SETMAG 1 2
TEXT 0 0 0 40 Font 0-0 at SETMAG 1 2
SETMAG 2 1
TEXT 0 0 0 80 Font 0-0 at SETMAG 2 1
SETMAG 2 2
TEXT 0 0 0 110 Font 0-0 at SETMAG 2 2
SETMAG 2 4
TEXT 0 0 0 145 Font 0-0 at SETMAG 2 4
; Restore default font sizes
SETMAG 0 0
FORM
PRINT
输出：
```

```
Font 0-0 at SETMAG 1 1
Font 0-0 at SETMAG 1 2
Font 0-0 at SETMAG 2 1
Font 0-0 at SETMAG 2 2
Font 0-0 at SETMAG 2 4
```

MULTILINE(ML)命令

使用MULTILINE（ML）命令可以以相同字体和行高来打印多行文本。

格式：

```
{command} {height}  
{text} {font} {size} {x} {y}  
{data}  
"  
  
{data}  
<ENDMULTILINE>
```

其中：

{command}: MULTILINE（ML）-打印多行文本。

{height}: 每一行文本的单位高度。

{text}: 文本命令（TEXT, VTEXT,等等）。

{font}: 字体的名称/编号。

{size}: 字体的大小标识。

{x}: 横向起始位置。

{y}: 纵向起始位置。

{data}: 要打印的文本。

<ENDMULTILINE> (or ENDML): 终止MULTILINE（命令）。

ML命令示例

输入：

```
! 0 200 200 210 1
```

```
ML 47
```

```
TEXT 4 0 10 20
```

```
1st line of text
```

```
2nd line of text
```

```
:
```

```
Nth line of text
```

```
ENDML
```

```
FORM
```

```
PRINT
```

输出：

```
1st line of text  
2nd line of text  
:  
Nth line of text
```

FONT-GROUP (FG) 命令

使用FG命令，用户可以将最多10个预先缩放的字体文件归类到一个组。用户以后可以在TTEXT命令中指定字体组。如果文本命令中使用了字体组，打印机将使用字体组中指定的最大字体，这将生成所需的文本数据，并且仍然保留在文本标签的可用宽度范围内。当在TEXT命令中进行指定时，{font}参数将指定为FG，{size}参数则指定为{fg}。请注意，用户还可以在COAT/ENCONCAT命令中指定FG命令。

格式：

```
{command} {fg fn fs} [fn fs] ...
```

其中：

{command}: FG

{fg}: 字体组编号。最多可指定 10 个字体组。有效字体组有效范围为 0-9。

{fn}: 字体名称/编号。

{fs}: 字体大小标识符。

备注：最多可以为字体组分配 10 个字号/字体大小对。

示例

输入：

```
! 0 200 200 250 1
; Specify fonts 0-0, 7-0, 5-0, 4-0 as members
; of font group 3.
FG 3 0 0 7 0 5 0 4 0
VT FG 3 10 250 Ketchup
VT FG 3 70 250 Fancy Ketchup
VT FG 3 120 250 Extra Fancy Ketchup
VT FG 3 180 250 Large Size Extra Fancy Ketchup
FORM
PRINT
```

输出：



在此示例中，文本将以指定字体组中的最大字体来打印，该字体组能够在250点标签字段中拟合所请求的文本。

4.条码指令

BARCODE 命令

BARCODE 命令能够以指定的宽度和高度纵向和横向打印条码。

标准条码

格式:

{command} {type} {width} {ratio} {height} {x} {y} {data}

其中:

{command}: 从下面选择一项:

BARCODE (或 B) : 横向打印条码。

VBARCODE (或 VB) 纵向打印条码。

{type}: 从下表中选择:

符号:	用法:
UPC-A	UPCA
UPC-E	UPCE
EAN/JAN-13	EAN13
EAN/JAN-8	EAN8
Code 39	39
Code 93/Ext.93	93
Code 128 (自动)	128
Codabar	CODABAR

备注: 条码数据必须在 {data}部分提供, 且应位于新的行字符序列之前。否则, 打印机可能会将下一条命令识别为条码数据, 因而生生成错误条码, 并导致下一条命令的执行错误。

{width}: 窄条的单位宽度。

{ratio}: 宽条与窄条的比率。

0 = 1.5 :1	20 = 2.0:1	26 = 2.6:1
1 = 2.0 :1	21 = 2.1:1	27 = 2.7:1
2 = 2.5 :1	22 = 2.2:1	28 = 2.8:1
3 = 3.0 :1	23 = 2.3:1	29 = 2.9:1
4 = 3.5 :1	24 = 2.4:1	30 = 3.0:1
	25 = 2.5:1	

备注: 附录中的比率为推荐值, 旨在帮助您获得最佳效果; 当然, 您可以指定任意比率。

{height}: 条码的单位高度。

{x}: 横向起始位置。

{y}: 纵向起始位置。

{data}: 条码数据。

条码示例

输入:

! 0 200 200 210 1

BARCODE 128 1 1 50 150 10 HORIZ.

TEXT 7 0 210 60 HORIZ.

VBARCODE 128 1 1 50 10 200 VERT.

VTEXT 7 0 60 140 VERT.

FORM

PRINT

输出: (如右图)



BARCODE-TEXT 命令

BARCODE-TEXT 命令用于通过创建条码时所用的相同数据来标记条码。这项命令避免了使用单独文本命令注释条码的必要。文本位于条码下方的中间位置。

使用 BARCODE-TEXT OFF（或 BT OFF）可以禁用它。

格式：

{command} {font number} {font size} {offset}

其中：

{command}: BARCODE-TEXT（或 BT）

{font number}: 注释条码时要使用的字体号。

{font size}: 注释条码时要使用的字体大小。

{offset}: 文本距离条码的单位偏移量。

BARCODE-TEXT 示例

输入：

```
! 0 200 200 400 1
JOURNAL
CENTER
; Annotate bar codes using font 7 size 0
; and offset 5 dots from the bar code.
BARCODE-TEXT 7 0 5
BARCODE 128 1 1 50 0 20 123456789
VBARCODE 128 1 1 50 40 400 112233445
BARCODE-TEXT OFF
FORM
PRINT
```

输出：（如右图）

PDF417（便携式数据文件）

格式：

{command} {type} {x} {y} [XD n] [YD n] [C n] [S n]
{data}

<ENDPDF>

其中：

{command}: 从下面选择一项：

BARCODE（或 B）： 横向打印条码。

VBARCODE（或 VB）： 纵向打印条码。

{type}: PDF-417

{x}: 横向起始位置。

{y}: 纵向起始位置。

[XD n]: 最窄元素的单位宽度。范围介于 1 至 32 之间，默认值为 2。

[YD n]: 最窄元素的单位高度。范围介于 1 至 32 之间，默认值为 6。

[C n]: 要使用的列数。数据列不包括起始/终止字符和左/右指示符。范围介于 1 至 30 之间，默认值为 3。

[S n]: 安全级别，指示要检测和/或纠正的最大错误量。范围介于 0 至 8 之间，默认值为 1。

{data} 条码数据。

<ENDPDF>: 终止 PDF-417。

备注：BARCODE-TEXT 命令不能用于 PDF-417 条码类型。对于任何所需的可人工识读文本，必须使用 TEXT 命令单独输入，如下例所示。



PDF417 示例

输入:

! 0 200 200 210 1

B PDF-417 10 20 XD 3 YD 12 C 3 S 2

PDF Data

ABCDE12345

ENDPDF

T 4 0 10 120 PDF Data

T 4 0 10 170 ABCDE12345

FORM

PRINT

输出:



PDF Data
ABCDE12345

QR Code

格式:

{command} {type} {x} {y} [M n] [U n]

{data}

<ENDQR>

其中:

{command}: 从下面选择一项:

BARCODE (或 B) : 横向打印条码。

VBARCODE (或 VB) : 纵向打印条码。

{type}: QR

{x}: 横向起始位置。

{y}: 纵向起始位置。

[M n]: QR Code 规范编号。选项是 1 或 2。QR Code Model 1 是原始规范, 而 QR Code Model 2 则是该符号的经过增强后的形式。Model 2 提供了附加功能, 而且可以自动与 Model 1 进行区分。Model 2 为推荐规范, 是默认值。

[U n]: 模块的单位宽度/单位高度。

范围是 1 至 32。默认值为 6。

{data}: 提供生成 QR Code 所需的信息。请参见下面的示例。

{data} 除了包含实际的输入数据字符串外, 还包含一些模式选择符号。输入数据类型可以由打印机软件自动识别, 也可以通过手动方式设置。模式选择符号和实际数据之间有一个分隔符 (逗号)。

用于自动选择数据类型的数据字段格式:

<Error Correction Level><Mask No.><Data Input Mode (should be "A")>,<Data Character String>

纠错级别应为以下符号之一:

H - 极高可靠性级别（H 级）；

Q - 高可靠性级别（Q 级）；

M - 标准级别（M 级）；

L - 高密度级别（L 级）。

掩码号可能会省略，也可能具有一个值（介于 0 至 8 之间）：

无 - 软件自动选择掩码；

0 至 7 - 使用带有相应编号（0 至 7）的掩码；

8 - 无掩码。

用于手动选择数据类型的数据字段格式包含字符模式符号，采用如下格式：

<Error Correction Level><Mask No.><Data Input Mode (should be "M")>，

<Character Mode 1><Data Character String 1>，<Character Mode 2><Data Character String

2>，< : > : >，<Character Mode n><Data Character String n>

字符模式符号：

N - 数字；

A - 字母数字；

Bxxxx - 二进制，二进制模式包含由 2 字节 BCD 代码表示的数据字符的数量 (xxxx)。

K - Kanji

不同的数据字段（带有对应的字符模式符号）由逗号分隔。

如果输入模式设置为“自动”，则无法设置 0x80 至 0x9F 和 0xe0 至 0xFF 的二进制代码。

<ENDQR>：终止 QR Code。

数据字段格式设置示例

示例 1

Error Correction Level:Standard level <M>

Mask No.:<None>

Input mode:Automatic setting <A>

Data:QR Code

The {data} field presentation for generating a QR code under the conditions above:

MA,QR Code

示例 2

Error Correction Level:Ultra high reliability level <H>

Mask No.:<0>

Input mode:Manual setting <M>

Character Mode:Numeric mode <N>

Data:0123456789012345

The {data} field presentation:

H0M,N0123456789012345

示例 3

Error Correction Level:Standard level <M>

Mask:<None> (Automatic selection)

Input mode:Manual setting <M>

Character Mode:Alphanumeric mode <A>

Data:AC-42

The {data} field presentation:

MM,AAC-42

示例 4

Error Correction Level:High density level <L>

Mask No.:Automatic setting <None>

Input mode:Manual setting <M>

Character Mode:Alphanumeric <A>

Data:QR code
 Character Mode:Numeric <N>
 Data:0123456789012345
 Character Mode:Alphanumeric <A>
 Data:QRCODE
 Character Mode:Binary

Data: qrcode

The {data} field presentation:

LM,AQRcode,N0123456789012345,AQRCODE,B0006qrcode

备注： BARCODE-TEXT 命令不能用于 QR Code。对于任何所需的可人工识读文本， 必须使用 TEXT 命令单独输入， 如下例所示。

QR Code 示例

输入：

! 0 200 200 500 1

B QR 10 100 M 2 U 10

MA,QR code ABC123

ENDQR

T 4 0 10 400 QR code ABC123

FORM

PRINT

输出：



QR code ABC123

备注： 可人工识读的文本不包含在 QR 代码输出结果中。

5.图形

BOX 命令

用户可以使用 BOX 命令生成具有指定线条宽度的矩形。

格式:

```
{command} {x0} {y0} {x1} {y1} {width}
```

其中:

{command}: BOX

{x0}: 左上角的 X 坐标。

{y0}: 左上角的 Y 坐标。

{x1}: 右下角的 X 坐标。

{y1}: 右下角的 Y 坐标。

{width}: 形成矩形框的线条的单位宽度。

BOX 命令示例

输入:

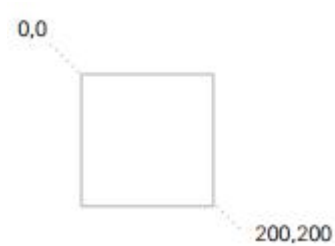
```
! 0 200 200 210 1
```

```
BOX 0 0 200 200 1
```

```
FORM
```

```
PRINT
```

输出:



备注: 输出中显示的文本坐标仅用于说明之目的。

LINE 命令

使用 LINE 命令可以绘制任何长度、宽度和角度方向的线条。

格式:

```
{command} {x0} {y0} {x1} {y1} {width}
```

其中:

{command}: 从下面选择一项:

LINE (或 L): 打印线条。

{x0}: 左上角的 X 坐标。

{y0}: 左上角的 Y 坐标。

{x1}: 以下项的 X 坐标:

- 水平轴的右上角。
- 垂直轴的左下角。

{y1}: 以下项的 Y 坐标:

- 水平轴的右上角。
- 垂直轴的左下角。

{width}: 线条的单位宽度

输入:

```
! 0 200 200 210 1
```

```

LINE 0 0 200 0 1
LINE 0 0 200 200 2
LINE 0 0 0 200 3
FORM
PRINT
输出:

```



备注：输出中显示的文本坐标仅用于说明之目的。

INVERSE-LINE 命令

INVERSE-LINE 命令的语法与 LINE 命令相同。位于 INVERSE-LINE 命令所定义区域内的以前创建的对象（包括文本、条码和/或图形（包括下载的.pcx 文件））的黑色区域将重绘为白色，白色区域将重绘为黑色。这些对象可以包括文本、条码和/或图形（包括下载的.pcx 文件）。INVERSE-LINE 对在其之后创建的对象不起作用，即使这些对象位于该命令的覆盖区域内也是如此。在示例 INVERSE2.LBL 中，在 INVERSE-LINE 命令之后创建的文本字段部分仍然为黑色，因此不可见，即使被放置在 INVERSE-LINE 区域内也是如此。

格式：

```
{command} {x0} {y0} {x1} {y1} {width}
```

其中：

{command}： 从下面选择一项：

INVERSE-LINE （或 IL）： 在现有字段上方打印一个线条以反转图像。

{x0}： 左上角的 X 坐标。

{y0}： 左上角的 Y 坐标。

{x1}： 以下项的 X 坐标：

- 水平轴的右上角。
- 垂直轴的左下角。

{y1}： 以下项的 Y 坐标：

- 水平轴的右上角。
- 垂直轴的左下角。

{width}： 反转线的单位宽度。

INVERSE-LINE 命令示例

输入 1：

```

! 0 200 200 210 1
CENTER
TEXT 4 0 0 45 SAVE
TEXT 4 0 0 95 MORE
INVERSE-LINE 0 45 145 45 45
INVERSE-LINE 0 95 145 95 45
FORM
PRINT
输出 1:

```


PCX命令

PCX命令使用户能够向打印机发送“.pcx”图形格式的图像。必须将.pcx图像编码为黑白图像。

格式:

{command} {x} {y}

{data}

其中:

{command}: PCX

{x}: 左上角的X坐标。

{y}: 左上角的Y坐标。

{data}: PCX 图像数据。

PCX 命令示例 1

在下面的示例中，图像分三个步骤发送。首先，打印机发送命令，指示将要发送.pcx格式的文件。其次，向打印机输入.pcx图像。此图像必须是双色（黑白）图像。最后，告知打印机打印标签。

输入 1:

```
! 0 200 200 500 1
```

```
PCX 0 30
```

```
Input 2 (IMAGE.PCX)
```

```
Input 3 (ENDPCX.LBL)
```

```
FORM
```

```
PRINT
```

输出:



PCX 命令示例 2

在本例中，PCX 映像已加载到打印机的闪存文件系统中，并命名为“IMAGE.PCX”。现在可以使用“!<”操作符指示打印机获取存储在“Image.PCX”文件中的数据，并将其用于构建映像。

输入2:

```
! 0 200 200 500 1
```

```
PCX 0 30 !<IMAGE.PCX
```

```
FORM
```

```
PRINT
```

输出2:



PATTERN 命令

PATTERN 命令与 LINE 和 SCALE-TEXT 命令一起使用，用于更改填充这些形状的模式。有效的模式值如下所示。

格式：

{command} {pattern number}

其中：

{command}: PATTERN

{pattern number}: 从下面选择一项：

100 填满（实心黑色/默认图案）。

101 水平线。

102 垂直线。

103 向右上升的对角线。

104 向左上升的对角线。

105 正方形图案。

106 剖面线图案。

输入：

```
! 0 200 200 700 1
```

```
; Draw horizontal and vertical patterns
```

```
PATTERN 101
```

```
LINE 10 10 160 10 42
```

```
PATTERN 102
```

```
LINE 170 10 350 10 42
```

```
; Draw left and right diagonal patterns
```

```
PATTERN 103
```

```
LINE 10 65 160 65 40
```

```
PATTERN 104
```

```
LINE 170 65 350 65 40
```

```
; Draw square and cross hatch patterns
```

```
PATTERN 105
```

```
LINE 10 115 160 115 40
```

```
PATTERN 106
```

```
LINE 170 115 350 115 40
```

```
; Draw a scalable text character with cross hatch pattern
```

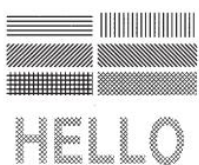
```
PATTERN 106
```

```
ST PLB_LAT.CSF 40 40 20 180 HELLO
```

```
FORM
```

```
PRINT
```

输出：



6. 高级命令

JUSTIFICATION 命令

使用对齐命令可以控制字段的对齐方式。默认情况下，打印机将左对齐所有字段。对齐命令将对所有后续字段保持有效，直至指定了其他对齐命令。

格式：

{command} [end]

其中：

{command}： 从下面选择一项：

CENTER： 居中对齐所有后续字段。

LEFT： 左对齐所有后续字段。

RIGHT： 右对齐所有后续字段。

[end]： 对齐的结束点。如果未输入参数，则对于横向打印，对齐命令将使用打印头的宽度；而对于纵向打印，对齐命令将使用零（页头）。

对齐示例

输入：

! 0 200 200 210 1

CENTER 383

TEXT 4 0 0 75 C

LEFT

TEXT 4 0 0 75 L

RIGHT 383

TEXT 4 0 0 75 R

FORM

PRINT

输出：

L C R

PAGE-WIDTH 命令

打印机假定页面宽度为打印机的完整宽度。打印会话的最大高度由页面宽度和可用打印内存决定。如果页面宽度小于打印机的完整宽度，则用户可以通过指定页面宽度来增加最大页面高度。

备注： 此命令应在打印会话开始时发出。

格式：

{command} {width}

其中：

{command}： 从下面选择一项：

PAGE-WIDTH（或 PW）： 指定页面宽度。

{width}： 页面的单位宽度。

PAGE-WIDTH 示例

输入 1：

! UTILITIES

SETLP 7 0 15

PW 300

PRINT

在打印此文本时，标签内存宽度设置为 300 点。

输出 1:

```
This text is printed with
label memory width set t
o 300 dots.
```

PAGE-HEIGHT命令

如果票据的大小应保持不变，并且纸张没有黑块来表示表单顶部，请使用PAGE-HEIGHT或PH命令。打印机将会把发送的数据划分为固定的页面大小。

PH 命令示例

输入:

```
! U1 PH {unit height}
```

PREFEED 命令

PREFEED 命令指示打印机在打印之前将介质向前移动指定长度。

备注：如果使用负值指定预送纸量，则该值不得超过标签的高度。这样做有可能使打印机陷入一个连续的循环之中并禁止与打印机进一步交互，直到关闭电源重新开机为止。

格式:

```
{command} {length}
```

其中:

{command}: PREFEED

{length}: 打印机在打印之前将介质向前移动的单位长度。

PREFEED命令示例

以下示例设置打印机在打印之前向前移动40个点行。

输入:

```
! 0 200 200 210 1
```

```
PREFEED 40
```

```
TEXT 7 0 0 20 PREFEED EXAMPLE
```

```
FORM
```

```
PRINT
```

PACE 命令

此命令可以与批量打印一起使用。在激活 PACE 后，用户必须按下打印机的 FEED（送纸）键才能打印其他标签，直至完成批次数。默认情况下，开机时定步功能处于禁用状态。

格式:

```
{command}
```

其中:

{command}: PACE

PACE 命令示例

在以下示例中，所显示的命令文件将一次性发送至打印机。对于其他两个打印输出，依次按下 FEED（送纸）键可分别产生一个打印输出。

输入:

```
! 0 200 200 210 3
```

```
; Tell printer to print a label
```

```
; after each ' FEED' key press
; until all 3 labels are printed
PACE
; Printer holds journal stock
JOURNAL
; Center the text
CENTER
TEXT 4 1 0 10 Print 3 labels
TEXT 4 1 0 90 Using PACE
PRINT
输出：
```



WAIT 命令

此命令用于在打印一个标签后引入一段延迟。

格式：

```
{command} {delay-time}
```

其中：

{command}: WAIT

{delay-time}: 延迟时间以 1/8 秒为单位。

WAIT 命令示例

在下例中，打印机将在打印每个标签后暂停 10 秒（ $10 * 8 = 80$ ）。

输入：

```
! 0 200 200 150 5
WAIT 80
TEXT 5 0 0 20 DELAY 10 SECONDS
FORM
PRINT
```

PRESENT-AT 命令

PRESENT-AT 命令可用于将介质定位在打印机的撕纸刀位置或操作员可轻松撕下标签的位置。当发出“PRESENT-AT”命令时，打印机将打印标签，然后在延迟一段时间后，将介质向前移动指定的距离。在开始一个新的打印作业之前，打印机会将介质回退相同的距离。

“delay”参数用于在执行批量打印作业时，避免不必要的向前/回退操作。PRESENT-AT 命令可以在标签文件中发出，也可在实用工具命令会话（! UTILITIES...PRINT）中发出。

注意：在使用此命令时，应对标签前缘和后缘应用18点的额外缓冲区。在介质上的任何预印图形上套印要打印的文件因标签而异。

格式：

{command} {length} {delay}

其中：

{command}: PRESENT-AT

{length}: 打印之后介质向前移动以及打印下一标签之前介质回退的单位长度（以点行为单位）。

{delay}: 打印标签后打印机在向前移动介质之前的等待时间间隔。以1/8秒为单位递增。延迟“1”等价于1/8秒。延迟“4”等价于1/2秒，以此类推。

PRESENT-AT命令示例

以下示例指示打印机等待1/4秒，如果在此期间打印机没有运行，则将介质向前移动80个点行。打印机将在打印下一个标签之前将介质回退相同距离。

输入：

```
! 0 200 200 250 1
```

```
TEXT 7 0 0 10 PRESENT-AT EXAMPLE
```

```
PRESENT-AT 80 2
```

```
FORM
```

```
PRIN
```

SPEED 命令

此命令用于设置电机的最高速度级别。每一款打印机型号都设置了最低和最高极限速度。SPEED 命令可以在 0 到 5 的范围内选择速度级别，0 表示最低速度。为每一款打印机型号设置的最高速度仅可在理想条件下达到。电池或供电电压、材料厚度、打印黑度、是否使用贴标机、是否使用剥离器以及标签长度等诸多因素均会影响最大极限打印速度。

警告：在练习此命令时，用户将会覆盖标签打印的出厂设定速度，这可能会导致打印质量下降。如果使用当前 SPEED 设置影响到打印质量，则应降低打印速度。

格式：

{command} {speed level}

其中：

{command}: SPEED

{speed level}: 一个介于 0 到 5 之间的数字，0 表示最低速度。

SPEED 命令示例

输入：

```
! 0 200 200 150 1
```

```
SPEED 4
```

```
TEXT 5 0 0 20 PRINTS AT SPEED 4
```

```
FORM
```

```
PRINT
```

SETSP 命令

SETSP 命令用于更改文本字符之间的间距。

格式:

{command} {spacing}

其中:

{command}: SETSP

{spacing}: 字符间的单位尺寸。间距的默认值为零。请注意, 此命令受单位命令设置的影响。

SETSP 命令示例

输入:

```
! 0 200 200 210 1
T 4 0 0 10 Normal Spacing
SETSP 5
T 4 0 0 50 Spread Spacing
SETSP 0
T 4 0 0 90 Normal Spacing
FORM
```

输出:



```
Normal Spacing
Spread Spacing
Normal Spacing
```

AUTO-PACE 命令

该命令可用于指示配备有取纸传感器的打印机延迟打印, 直到清除之前打印的标签。

格式:

{command}

其中:

{command}: AUTO-PACE

AUTO-PACE 命令示例

此示例指示打印机打印 10 个标签。打印机打印一个标签, 然后等待该标签被清除, 接着打印下一个标签。

输入:

```
! 0 200 200 250 10
CENTER
TEXT 7 0 0 10 AUTO-PACE EXAMPLE
AUTO-PACE
FORM
PRINT
```

ON-FEED 命令

当按下 FEED 走纸键或打印机收换页符（0x0c）时，打印机可以配置为忽略、换页、或重新打印上一个标签。

格式：

{command} {action}

其中：

{command}: ON-FEED

{action}: 从下面选择一项：

IGNORE: 当按下走纸键或接收到换页字符时，不采取任何操作。

FEED: 当按下走纸键或接收到换页字符时，切换至下一页顶部。

REPRINT: 当按下走纸键或接收到换页字符时，重新打印上一个标签。

在以下示例中，所示的命令文件仅发送到打印机一次。对于其他两个标签，依次按下 FEED 走纸键可分别打印一个标签。

ON_FEED命令示例

输入：

! 0 200 200 300 1

ON-FEED REPRINT

CENTER

JOURNAL

TEXT 4 1 0 20 PRESS FEED KEY

TEXT 4 1 0 100 TO REPRINT

TEXT 4 1 0 180 THIS TEXT

PRINT

输出：

PRESS FEED KEY
TO REPRINT
THIS TEXT

PRESS FEED KEY
TO REPRINT
THIS TEXT

PRESS FEED KEY
TO REPRINT
THIS TEXT

COUNTRY/CODE PAGE 命令

使用 COUNTRY 控制命令可以针对指定的国家/地区替代适当的字符集。

格式:

{command} {name}

其中:

{command}: COUNTRY

{name}: 从下表中选择:

USA SWEDEN ITALY LATIN9 KOREA (韩语, 双字节字符集, 参见第 8-23 页上的“亚洲字体”主题)	GERMANY SPAIN CP850 CP874 (泰语) BIG5 (繁体中文, 双字节字符集, 参见第 8-23 页上的“亚洲字体”主题)	FRANCE NORWAY UK CHINA (简体中文, 双字节字符集, 参见第 8-23 页上的“亚洲字体”主题) JAPAN-S (S-JIS, 双字节字符集, 参见第 8-23 页上的“亚洲字体”主题)
--	--	---

COUNTRY 命令示例

输入:

```
! 0 200 200 80 1
IN-MILLIMETERS
JOURNAL
CENTER
; Set the country as USA
COUNTRY USA
; Now Print Text From ISO substitution Table
TEXT 4 0 0 8 COUNTRY IS USA
TEXT 4 0 0 15 #${@[\]^'{}~}~
; Set country for France and print the same text
COUNTRY FRANCE
TEXT 4 0 0 28 COUNTRY IS FRANCE
TEXT 4 0 0 35 #${@[\]^'{}~}~
PRINT
```

输出:

```
COUNTRY IS USA
#${@[\]^'{}~}~

COUNTRY IS FRANCE
£$à°ç$~µéùè"
```

DEFINE FORMAT

可以通过以下方式定义标签格式文件：使用 **DEFINE-FORMAT**（或 **DF**）命令标记格式开始位置，使用 **PRINT** 标记结束位置。使用“\\”（双反斜杠）作为数据的占位符。

DEFINE FORMAT 命令示例

输入：

```
! DF SHELF.FMT
! 0 200 200 210 1
CENTER
TEXT 4 3 0 15 \\
TEXT 4 0 0 95 \\
BARCODE UPCA 1 1 40 0 145 \\
TEXT 7 0 0 185 \\
FORM
PRINT
```

USE-FORMAT

USE-FORMAT（或 **UF**）命令指示打印机使用指定的格式文件。将使用该格式文件以及 **USE-FORMAT** 命令后面提供的数据创建标签。在访问指定的格式文件后，打印机将使用所提供的数据替代“\\”分隔符，从而生成所需的标签。

USE FORMAT 命令示例

输入：

```
! UF SHELF.FMT
$22.99
SWEATSHIRT
40123456784
40123456784
```

与所有打印命令一样，格式文件的每一行及其伴随的变量必须以回车符和换行符序列结尾。

在定义之后，格式将保留在打印机的非易失性内存中以供以后引用。可以通过重写格式文件来更改现有格式。通过使用 **DEL** 命令，可以删除格式文件。

格式文件名可由不超过 8 个的字母或数字组成，格式文件扩展名可由不超过 3 个的字母或数字组成。

格式文件名或扩展名中的所有小写字母将转换为大写字母。

备注：例如，每次在打印机上使用“**! DEFINE-FORMAT..**”、“**! DF..**”或 **Label Vista** 应用程序创建文件时，文件信息将写入闪存中。与 **RAM** 不同，闪存不需要电池来保留数据，也不会因静电放电而导致数据损坏。尽管闪存存在文件内容的安全保护方面要优于 **RAM**，但它受到平均 10,000 次写周期（即，文件创建）的限制。出于此原因，用户在练习使用文件创建命令时应注意不要超过所声明的次数限制。

BEEP 命令

此命令用于指示打印机让蜂鸣器发出给定时间长度的声音。未配备蜂鸣器的打印机将忽略此命令。

格式:

{command} {beep _ length}

其中:

{command}: BEEP

{beep _ length}: 蜂鸣持续时间, 以 1/8 秒为单位递增指定。

BEEP 命令示例

此示例指示打印机蜂鸣两秒钟 (16 x .125 秒 = 2 秒)

输入:

```
! 0 200 200 210 1
```

```
CENTER
```

```
TEXT 5 0 0 10 beeps for two seconds
```

```
BEEP 16
```

```
FORM
```

```
PRINT
```


7.行式打印模式

SETLP 命令

选择行式打印机字体（SETLP 命令）将会更改打印机用于行式打印模式的字体。它还会选择打印机收到回车符（十六进制值 0x0d）时打印机向下移动的空间量。

! U1 SETLP {font name or number} {size} {unit height}

{unit height} 值应设置为所用字体的实际高度。

SETLP 允许您使用常驻字体或下载到闪存中的预缩放字体。Label Vista 设计软件可以基于任何可用的 TrueType1 字体为打印机创建和上传字体。

在使用行式打印机制作收据时，可以多次设置打印机字体。例如，若要使用大号字体将公司名称放置在标签顶部，可以更改为字体 5 大小 2，然后再更改为字体 7 大小 0。

SETLP 命令示例

输入：

! U1 SETLP 5 2 46

AURORA'S FABRIC SHOP

! U1 SETLP 7 0 24

123 Castle Drive, Kingston, RI 02881

(401) 555-4CUT

输出：

AURORA'S FABRIC SHOP
123 Castle Drive, Kingston, RI 02881
(401) 555-4CUT

SETLF 命令

使用 SETLF 命令可在不更改字体的情况下更改每行的高度。

格式：

! U1 SETLF {unit height}

命令“! U1 SETLF 40”可以针对收到的每个 LF（换行，十六进制值 0x0a）字符，将纸张向前走 40 个点。

SETLF 命令示例

输入：

! U SETLP 4 0 40

SETLF 40

PRINT

Output 2

文本行

文本行

文本行

输出：

Output 2

Text line

Text line

Text line

使用 X 和 Y 坐标移动

尽管打印机处于行式打印模式，但仍可以使用 X 和 Y 值将纸张纵向和横向移动。

格式：

```
! U1 X {unit value}
! U1 Y {unit value}
! U1 RX {unit x value to move relative to present position}
! U1 RY {unit y value to move relative to present position}
```

此命令用于在纸张上横向移动而不使用额外空格，或者在纸张上纵向移动而无需将 SETLF 命令设置为特定值。

“Y”坐标不能使用负值。

LMARGIN 命令

LMARGIN 命令用于设置行式打印模式中的左边距。LMARGIN 命令不会发出多个 X 命令或插入空格，而是将所有内容移动所选的点数。

格式：

```
! U1 LMARGIN {dots to offset from left}
```

此函数可与 PAGE-WIDTH 命令一起使用。LMARGIN 会将左边距从自动计算的纸张边缘移动设定的点数。

SETBOLD 命令

SETBOLD 命令可使文本加粗并稍微加宽。SETBOLD 命令会采用一个操作数来设置文本变黑的程度。

格式：

```
! U1 SETBOLD {value}
```

其中，{value} 是介于 0 到 5 之间的偏移量。

备注：value 将采用通过单位命令设置的单位。

默认单位设置以点为单位。（203 点 = 1 英寸）

如果单位为英寸，则偏移值的范围为 0-0.0246 英寸。

如果单位为厘米，则偏移值的范围为 0-0.0625 厘米。

如果单位为毫米，则偏移值的范围为 0-0.625 毫米。

完成后，请务必发出“! U1 SETBOLD 0”命令以禁用粗体格式。

SET BOLD 命令示例

输入：

```
! U1 SETBOLD 2
```

```
This text is in bold ! U1 SETBOLD 0
```

```
but this text is normal.
```

输出： 

LP-ORIENT 命令

LP-ORIENT 命令设置将要打印的行式打印字符的旋转角度。

格式：

```
{command} {value}
```

其中：

{command}: LP-ORIENT

{value}: 选择旋转角度

0 (默认)

270

备注：采用这两种旋转角度时，将按字符的发送顺序打印字符。

SETFF 命令

SETFF 命令用于将介质的顶部与打印头对齐。执行该命令后，将在以下情况下进行对齐：

- 按下 FEED（送纸）键。
- 发出换页符 (0x0c)。
- 发出 FORM 命令。

格式：

<!> <UTILITIES>

{command} {max-feed} {skip-length}

<PRINT>

其中：

{command}: SETFF

{max-feed}: 打印机在搜索下一个可视标记以对齐页面顶部时向前走纸的最大单位长度。有效值为 0-20,000。

{skip-length}: 打印机越过页面顶部向前走纸的单位长度。有效值为 5-50。

SETFF 命令示例

以下示例对打印机进行编程，以便在找到可视标记之前或者在走纸距离达到最大值 25 毫米之前一直向前走纸。如果找到可视标记，纸张将再向前走 2.5 毫米。

输入：

! UTILITIES

IN-MILLIMETERS

SETFF 25 2.5

PRINT

8. 高级实用工具

TIMEOUT 实用工具

使用 TIMEOUT 命令可设置打印机在不接收数据时保持开机的时间。如果在指定超时后未接收任何数据，打印机将自行关闭以节省电量并延长电池寿命。您可以通过将超时值设置为 0 来禁用超时功能。

格式：

```
<!> <UTILITIES>
{command} {time}
<PRINT>
```

其中：

{command}: TIMEOUT

{time}: 打印机自行关闭前无活动的时间（以 1/8 秒为单位）。

TIMEOUT 实用工具示例

输入：

```
! UTILITIES
TIMEOUT 960
PRINT
```

本例将打印机设置为在无活动 2 分钟后关闭（120 秒 X 8 = 960）。

CHECKSUM 实用工具

此命令用于以四位字符组成的以 Null 终止的 ASCII 字符串形式报告应用程序校验和。

格式：

```
<!> <UTILITIES>
{command}
<PRINT>
```

其中：

{command}: CHECKSUM

CHECKSUM 示例

输入：

```
! UTILITIES
CHECKSUM
PRINT
```

DIR 实用工具

DIR 命令用于发送文件目录给主机。

格式：

```
<!> <UTILITIES>
{command}
<PRINT>
```

其中：

{command}: DIR

DEL 实用工具

DEL 命令用于删除指定的文件。

格式:

<!> <UTILITIES>

{command} {name.ext}

<PRINT>

其中:

{command}: DEL

{name.ext}: 要删除的文件名称。

备注: DEL *.*可用于全局删除所有文件。

DEFINE-FILE (DF) 实用工具

DF 命令用于定义要加载到打印机中的文件的名称。如果打印机中存在同一名称的文件, 它将被新文件覆盖。文件的内容必须包含 ASCII 字符。要将二进制文件传输到打印机, 请使用 Label Vista 应用程序中提供的实用工具。

格式:

{data}

{terminator}

其中:

{command}: DF

{filename.ext}: 要创建的文件的名称。

{data}: 文件的内容。文件必须为ASCII形式且不能包含任何{terminator}关键字。

{terminator}: 从下面选择一项:

PRINT: 如果使用了PRINT终止符, 它也会被写入文件中。

END: 如果使用了END终止符, 它不会被写入文件中。

DEFINE-FILE 示例

输入:

! DF AUTOEXEC.BAT

! UTILITIES

SETFF 200 20

PRINT

备注: 每当使用"! DEFINE-FORMAT...", "!DF..."或LabelVista等应用程序在打印机上创建文件时, 都会将文件信息写入内存。与RAM不同, 闪存保留数据时不需要电池, 并且可以通过静电放电保证数据不受损坏。尽管闪存在安全保护文件内容方面优于RAM, 但它平均只能支持10,000个写入循环(即文件创建)。用户应调用文件创建命令, 以便不超出写入循环次数。

TYPE 实用工具

TYPE命令可通过从打印机发送文本文件到主机来读取它。

格式:

<!> <UTILITIES>

{command} {name.ext}

<PRINT>

其中:

{command}: TYPE

{name.ext}: 发送到主机的文本文件名称。

GET-TIME 实用工具

此命令用于以八位字符组成的以Null终止的ASCII字符串形式报告当前时间（如果有效）。未配备实时时钟模块的打印机将忽略该命令。

格式:

<!> <UTILITIES>

{command}

其中:

{command}: GET-TIME

打印机输出: hh:mm:ss\0

hh = 小时 (00 – 23)

mm = 分钟 (00 – 59)

ss = 秒 (00 – 59)

\0 = Null终止符 (00H)

GET-DATE 实用工具

此命令用于以8个字符组成的以Null终止的ASCII字符串形式来报告当前日期（如果有效）。没有配备实时时钟模块的打印机将忽略此命令。

格式:

<!> <UTILITIES>

{command}

<PRINT>

其中:

{command}: GET-DATE

打印机输出: mm:dd:yyyy\0

mm = 月 (01 – 12)

dd = 日 (01 – 31)

yy = 年 (1990 – 2089)

\0 = 空终止符 (00H)